

Self-Reforming Routing for Stochastic Search in VLSI Interconnection Layout

Yukiko KUBO Yasuhiro TAKASHIMA[†] Shigetoshi NAKATAKE[‡] Yoji KAJITANI

Department of
Electrical and Electronic Eng.
Tokyo Inst. of Technology
2-12-1 Ookayama, Meguro-ku,
Tokyo 152-8552, Japan
Tel: +81-3-5734-2665
Fax: +82-3-5734-2902

[†] Information Science
Japan Advanced Inst. of
Science and Technology
Tatsunokuchi, Nomi-gun,
Ishikawa 932-1292, Japan
Tel: +81-761-51-1284
Fax: +81-761-51-1284

[‡] Faculty of International Environmental Eng.
Promotion and Development Office
Kitakyushu University
Kokuraminami-ku, Kitakyushu-city
Fukuoka 802-8577, Japan
Tel: +81-93-964-4270
Fax: +81-93-964-4275

e-mail: kubo@ss.titech.ac.jp takasima@jaist.ac.jp nakatake@kitakyu-u.ac.jp kajitani@ss.titech.ac.jp

Abstract— Given a route which connects terminals on a one-layer routing area (Steiner tree), flip is a procedure that makes a current route change its configuration within its peripheral domain. A flip reforms a route by replacing one of its edges with a minimal detour. A route can flip one nearby obstacle. If the obstacle is another route, a more organized operation, called the dual flip, is applied to a pair of routes. The idea is enhanced to the 2-layer hv-routing. The performance of flip and dual flip was experimented in simulated annealing which reforms a route or a set of routes with respect to the evaluation function of multiple objectives. Some unique and satisfiable results were observed.

I. INTRODUCTION

Recent development of interconnection technologies enables automation tools to complete routing without much difficulties. !!! However, it does not mean at all any solution to the difficulties due to the design environment caused by, for example, 3D-capacity, crosstalk, inductance, non-linear delay model, and other miscellanea that aggravate the yield.

Simultaneously, it is being understood that the constructive ways of routing that takes these multiple performance targets into considerations cannot be effective. There have been many studies that are successful as a single performance driven routing, such as [2, 4, 5, 6, 7]. But their enhanced algorithms to pursue multiple objectives are not convincing enough because the weight of different objectives is determined not based on a concrete base. As results, the routing for performance has become a more serious problem than ever.

Owing to the recent progress of computation resources, stochastic search is becoming effective. If we use this methodology, the key issue is in a quick generation of

feasible routes. For the purpose, we propose a procedure, called the flip. It makes one route change its configuration within its peripheral domain. Here a route is modeled by a Steiner tree on a planar graph connecting designated terminals. By flip, a route is reformed by replacing one of its edges with a minimal detour in the sense that it flips a single nearby obstacle. It is proved that any configuration is reachable from any configuration by a proper repetition of flips.

Some enhancements are proposed for practical use. If the obstacle is another route, a dynamical operation, called the dual-flip, is applied. For the 2-layer routing area by hv-rule, a similar operation is introduced.

An algorithm which improves the route configuration or a set of route configurations was implemented in a simulated annealing where the move is the flip or dual-flip. The evaluation was chosen from the total length, bend (via), skew, and crosstalk. The results were not only satisfiable as statics but also interesting.

Finally we note the difference of flip and well-known operation "rip-up and reroute" (abbreviated as R-R) which shows a similar behavior. See [1] for example. R-R is a step in construction stage: it tries to make an unrouted connection complete at the sacrifice of other completed routes. As its nature, R-R is triggered only when it meets an obstacle that blocks the routing. R-R cannot be used as a move in stochastic search. While, the flip reforms a route to adapt itself to the circumstances.

II. PRELIMINARIES

In our model, a connection requirement is defined on a plane graph $G(V, E)$ such that a given set of terminals $R \subset V$ shall be connected. For simplicity of discussions, we assume that a graph is *biconnected graph*, that is, any pair of vertices in the graph are connected by two or more vertex disjoint paths.

When we consider a graph which is not biconnected, we divide the graph into a set of biconnected components and consider each component independently.

The edge set that connects the terminals as specified is called a *Steiner edge set*. It allows any redundant edges to be included but it is called a *Steiner tree* if it is minimal i.e. deletion of any one edge makes it not a Steiner edge set.

Lemma 1 *A Steiner edge set is a Steiner tree if and only if it contains neither cycle nor a pendant edge. A pendant edge is an edge whose one end vertex is not a terminal but of degree 1.*

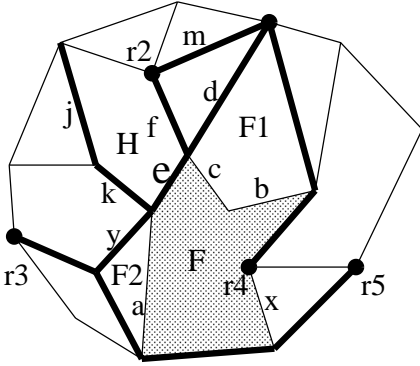


Fig. 1. Terminal set $R = \{r1, r2, r3, r4, r5\}$, a Steiner edge set S , pendant edge j , a Steiner tree $T = S - \{j, k, m\}$, and face F (shaded)

Given a Steiner edge set S , an operation to repeat deletion of a pendant edge or an edge in a cycle as long as a pendant edge or a cycle remains is called the *reduction* of a Steiner edge set S . In Fig.1, deletion of edges $\{j, k, m\}$, $\{j, k, d\}$ or $\{j, k, f\}$ is the reduction of S .

A cycle divides the plane into two regions. One is called the *inner* region if it is in the right-hand side traversing the cycle clock-wise, and the other the *outer* region.

Given a cycle, a path in the inner region that connects two distinct vertices on the cycle is called a *chord path*. A cycle without any chord path is called the *face-cycle*. The inner region of a face-cycle is the *face*. If a face is denoted by F , its corresponding face-cycle is denoted by \overline{F} .

An edge belongs to two face-cycles. The outer region of the graph is considered as a face. In the figure, e belongs two face-cycles \overline{F} and \overline{H} .

Lemma 2 *For a Steiner tree T and a face-cycle \overline{F} such that $e \in T \cap \overline{F}$,*

1. $T' = T \cup \overline{F} - \{e\}$ is a Steiner edge set, and
2. reduction of T' leads a unique Steiner tree if deletion of cycle edges is restricted to those on \overline{F} .

In Fig.1, given $T = S - \{j, k, m\}$, $T' = T \cup \overline{F} - \{e\}$ contains two face-cycles $F1, F2$. After the reduction as in the lemma, the resultant Steiner tree is uniquely $T' - \{a, b, c, y\}$.

III. FLIP

The flip is an operation which works for any 3-tuple of a Steiner tree, a face-cycle and an edge.

procedure **flip**(T, e, \overline{F})

step 1: If $e \notin T \cap \overline{F}$, output T . Otherwise:

step 2: $T = T \cup \overline{F} - \{e\}$

step 3: Apply the reduction to T where cycle edges to be deleted is restricted to those that belong to \overline{F} .

An example of one-time application of flip is shown in Fig.2.

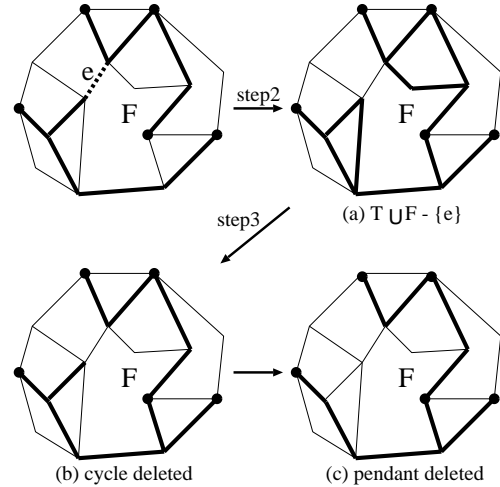


Fig. 2. Flip for input (T, e, \overline{F})

Theorem 1 *The flip(T, e, \overline{F}) outputs a unique Steiner tree.*

As a universal operation independent from the objective function, the reachability is crucial where a Steiner tree is said *reachable* from another Steiner tree if the former is obtained from the latter by repetition of proper flips.

Theorem 2 *Any Steiner tree is reachable from any Steiner tree by at most $|E| - |V|$ repetitions of flips.*

If we need a rigorous proof, we have to introduce some function which measures the difference between two Steiner trees. For the space, we sketch the proof taking an

example shown in Fig.3. We are going to apply necessary flips to get T_b from T_a .

Let B denote the sum of inner regions of cycles in $T_a \cup T_b$. In the figure, it is the shaded region. Let $d(T_a, T_b)$ be the number of distinct face-cycles contained in B . Note that $T_a = T_b$ if $d(T_a, T_b) = 0$. This function is used to indicate the degree of difference between two Steiner trees. In (a), $d(T_a, T_b) = 5$. Find an edge e such that $e \in T_b - T_a$ and let F be a face in B such that $e \in \overline{F}$. Then, it is trivial that after applying $\text{flip}(T_a, e, \overline{F})$, $d(T_a, T_b)$ is decreased. In the figure, T_a in (a) is transformed to T'_a in (b) by $\text{flip}(T_a, e, \overline{F})$ with $d(T'_a, T_b) = 4$. Again T'_a in (b) is transformed to T''_a as shown in (c) with $d(T''_a, T_b) = 2$. After two times applications of flip (neglected in the figure), $d(T_a, T_b)$ becomes zero, i.e. two Steiner trees are identical.

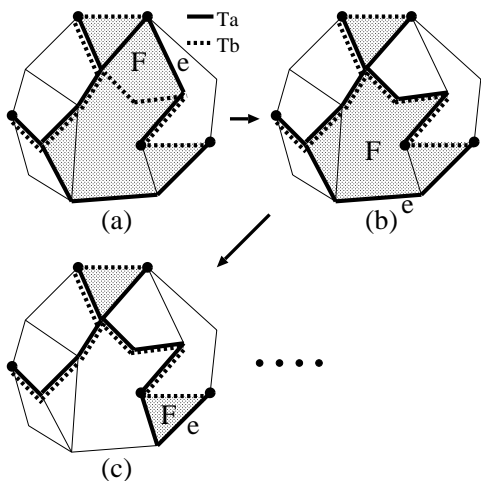


Fig. 3. Transformation of T_a to T_b by repetition of flips

IV. EXTENSION OF FLIP

The model on which flip is defined is too simple. We can extend the idea in two directions, one is to handle more than one routes and the other the 2-layer routing.

A. Plural Nets Optimization

The flip is an operation which focuses a single Steiner tree where other circuit elements are equally regarded as obstacles. However, in case when the obstacle is another route, it could be reformed if it is necessary for improvement of two Steiner trees. For the purpose, a new operation, the *dual-flip*, is introduced.

A rough description of the dual-flip for a pair (T_a, T_b) of Steiner trees is given as follows. See Fig. 4. First apply $\text{flip}(T_a, e_a \in T_a, \overline{F})$ to get new T'_a such that it has a cross on \overline{F} with T_b .

Then by choosing a proper edge of T_b and a face-cycle, apply $\text{flip}(T_b, e_b \in T_b, \overline{F})$, to get T'_b such that T'_a and T'_b

are released from crossing. This is possible if there is face-cycle \overline{F} that shares a path on T_a and that on T_b .

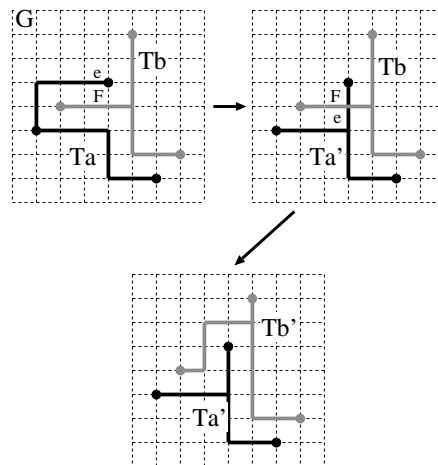


Fig. 4. Dual-flip: concatenation of flip to T_a and flip to T_b

B. General Routing Region

The flip has been introduced assuming that the routing region or its model graph is planar. However, the concept of flip is based on two procedures “disconnect by deleting an edge” and “reconnect by a detour”. Here the planarity is used only in the choice of the detour, or the face-cycle. Hence it is straightforward to extend the idea applicable to any region only if a way is fixed to choose the detour with respect to e .

The most important extension is to include 2-layer routing. It is easy, especially if the routing follows the hv-rule. The routing region graph is still represented by a planar grid, with an additional condition that horizontal path and vertical path can cross without fusing. Then, the minimal detour is naturally defined. An example shown in Fig.5, where deletion of edge e disconnects the Steiner tree (thick line). The connection is recovered by a path that has no crossing with the obstacles.

V. EXPERIMENTS

We implemented procedures flip and dual flip into the algorithm:

algorithm **Self-Reforming Routing**

Initial: Construct an initial routing by any existing algorithm (or assume this is input).

Process: Iterate

- Evaluate the solution, and keep the best so far.
- Update the solution by flip or dual-flip.

The algorithm is controlled by simulated annealing (SA), where parameters are decided very standard.

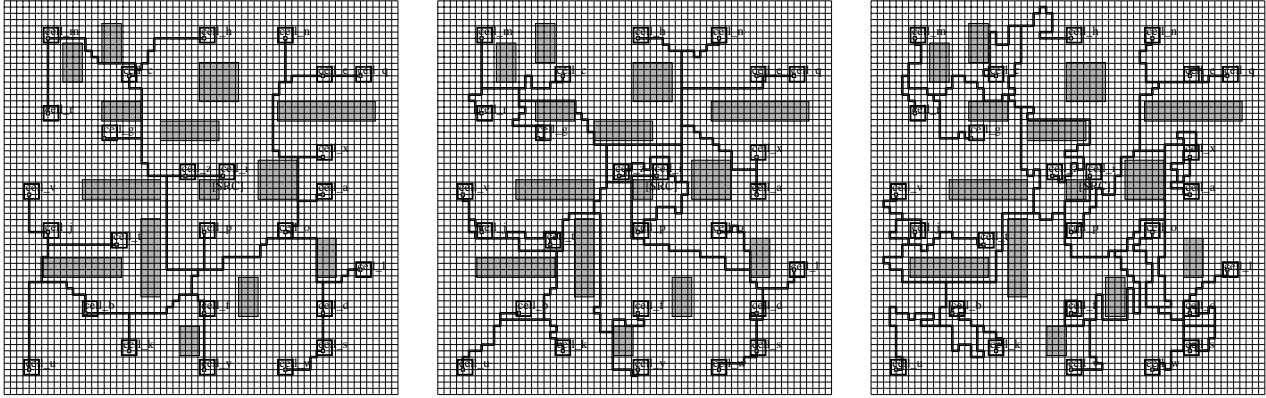


Fig. 6. Exp.A:Input tree with 25 sinks(left), a result for $\alpha = 0.25$ (center) and a result for $\alpha = 1$ (right)

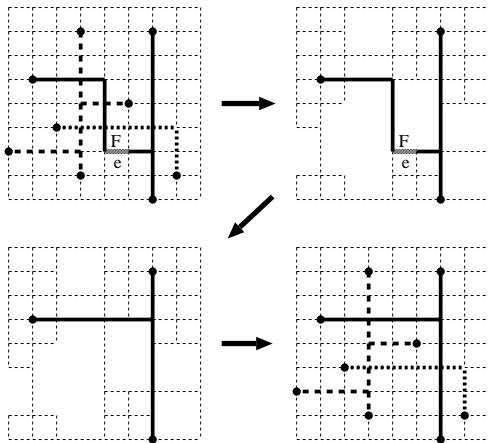


Fig. 5. 2-layer hv-routing: flip (T, e, \overline{F}) where the obstacle is another Steiner tree

A. Clock Trees with Skew and Length Concerned

The purpose of improving the clock tree is two folded: length and skew. We try to improve an input clock tree shown in Fig. 6(left). This is the one obtained by the Prim based maze routing. Shaded 14 rectangles in the figure are obstacles.

The move of SA is the flip. We set the evaluation function to $(LEN) + \alpha(SKEW)^2$, where LEN is the length of the Steiner tree, $SKEW$ is the maximal skew of the length from the source to the sinks, and α is a parameter coefficient. If α is small, our algorithm seeks a Steiner-tree with minimal length, while if α is large, it tries to achieve zero-skew routing.

Experiments are for three cases of $\alpha = 0, 0.25$, and 1. The resultant configurations are shown in the Fig.6(center) and (right). The data is given in TABLE. I.

Observation: From TABLE.I, we could say that the algorithm is very faithful to the objective function.

A nearly zero-skew wiring is attained when $\alpha = 1$. It must be noted that conventional clock-tree synthesis methods, which are H-tree method or Deferred-merge

	LEN	SKEW	time(sec)
input tree	1344	352	-
$\alpha = 0$	1184	272	363
$\alpha = 0.25$	1848	48	440
$\alpha = 1$	2728	4	529

TABLE I

EXP.A: WIRE LENGTH AND MAXIMAL SKEW OF A CLOCK TREE

method, are inherently unable to be applied when the region is not homogeneous by obstacles such as terminals, vias and other routes.

B. Plural Steiner Trees with Length Concerned

Suppose we are given five data each consisting of 4, 5, 6 routes as shown in TABLE.II. The configuration of input D3 is shown in Fig.7(left). The improvement request is to reduce the total length. We apply the flip and dual flip. The resultant trees are shown in Fig. 7 (center) with performance data in TABLE.III.

Observation: All the trees were equally significantly improved by around 20%.

C. Plural Steiner Trees with Multiple Objectives

Suppose we are given three data same as D1, D2 and D3 of Exp. B as shown in TABLE.IV. They are Steiner trees obtained through the Prim based maze routing. Hence they are the results obtained by single-performance driven constructive method.

The request is to minimize following four values; LEN, BEND(number of bends), SKEW, and CTLK (crosstalk, the sum of parallel running length of two routes of distinct Steiner trees).

Let the total evaluation function be their weighted sum with coefficients L, B, S, and C, respectively.

The output Steiner-trees of D3 by weighting C are shown in Fig. 7(right).

Observation: Several noticeable issues are listed.

	#net	#term	LEN
D1	6	15	443
D2	4	12	525
D3	5	13	559
D4	5	15	498
D5	5	16	510

TABLE II

EXP.B:INPUT DATA AND THE LENGTH OF INPUT TREES

	length		time(sec)	
	flip only	flip & dual	flip only	flip & dual
D1	422	376	76	69
D2	494	397	125	114
D3	495	425	97	92
D4	438	398	78	73
D5	543	448	118	106

TABLE III

EXP.B:RESULTS FOR LENGTH CONCERNED BY FLIP AND DUAL-FLIP

1. If no special weight is imposed to any performance, BEND, SKEW and CTLK are drastically improved. This is not surprising since no such attention has been given to the initial solution.
2. It is remarkable that even LEN can be improved. Remember that LEN of the input has been locally optimized.
3. The effect of weighting is apparent if we see the values on the diagonal of the table. This shows the potential of our self-reforming algorithm to respond the users.

D. Steiner Trees on 2-Layer with Crosstalk and Vias Concerned

The input is tens of routes on the 2-layer by hv-rule with respect to the crosstalk and the number of vias.

To see the average potential of our algorithm, we generate a number of input routes. One input is called *class-n* if it has *n* routes. We generates 10 instances for *class-30*, *class-40*, *class-50* each. One of the input is shown in Fig. 8 (left).

Before mentioning the result, some comments that may affect the results shall be added.

The input routes are the ones designed by the Prim based maze routing and then improved by reducing unnecessary vias. Though these algorithms are order dependent of the nets to be executed, we may say that they are already optimized, at least locally with respect to the length and the number of vias.

Furthermore, we checked out the route as infeasible if its length is more than 1.1 times of that of the input. This is from an experience of Experiment C, that if any length is allowed, to get a good route with respect to the crosstalk will be easy. However, such a long route is not

	LEN	BEND	SKEW	CTLK
D1	443	116	53	97
D2	525	129	57	81
D3	559	150	65	122

TABLE IV

EXP.C:INPUT DATA AND DATA OF INPUT TREE

	L:B:S:C	LEN	BEND	SKEW	CTLK
D1	1:1:1:1	454	80	41	11
	10:1:1:1	423	67	37	9
	1:10:1:1	442	59	53	127
	1:1:10:1	497	88	1	6
	1:1:1:10	508	97	1	0
D2	1:1:1:1	536	78	50	24
	10:1:1:1	523	72	57	31
	1:10:1:1	530	61	71	109
	1:1:10:1	642	148	15	34
	1:1:1:10	557	87	50	0
D3	1:1:1:1	678	99	1	56
	10:1:1:1	555	72	63	78
	1:10:1:1	559	72	63	78
	1:1:10:1	719	175	1	45
	1:1:1:10	698	119	1	1

TABLE V

EXP.C:EFFECTS OF WEIGHTING

class	weight 0:1		weight 1:9		weight 1:3	
	VIA	CTLK	VIA	CTLK	VIA	CTLK
class-30	3.75	0.36	2.14	0.39	1.78	0.53
class-40	3.41	0.56	2.10	0.61	1.40	0.79
class-50	2.91	0.73	1.87	0.74	1.14	0.94

TABLE VI

EXP.D:AVERAGE IMPROVEMENT OF ROUTES ON 2-LAYER

accepted as an improvement. We do not like to allow an improvement of crosstalk at the sacrifice of length.

The operation in SA is the flip.

The experiment was done three times, changing the evaluation function in the weight of the number of vias and crosstalk.

The result is given in TABLE. VI whose figures represent the ratio over the input. The result of the route given in Fig.8(left) is shown in the figure (center) and (right).

Observation: In average, in every class, the crosstalk was reduced considerably, apparently as a trade-off of the number of vias. This seems caused by the definition of the crosstalk that only exactly parallel running is counted.

VI. CONCLUSION

Discussing that the search based algorithm for performance improvement of the routes is only the plausible way in recent design environment and that the key in such a technology is in generation of the candidate routes, we

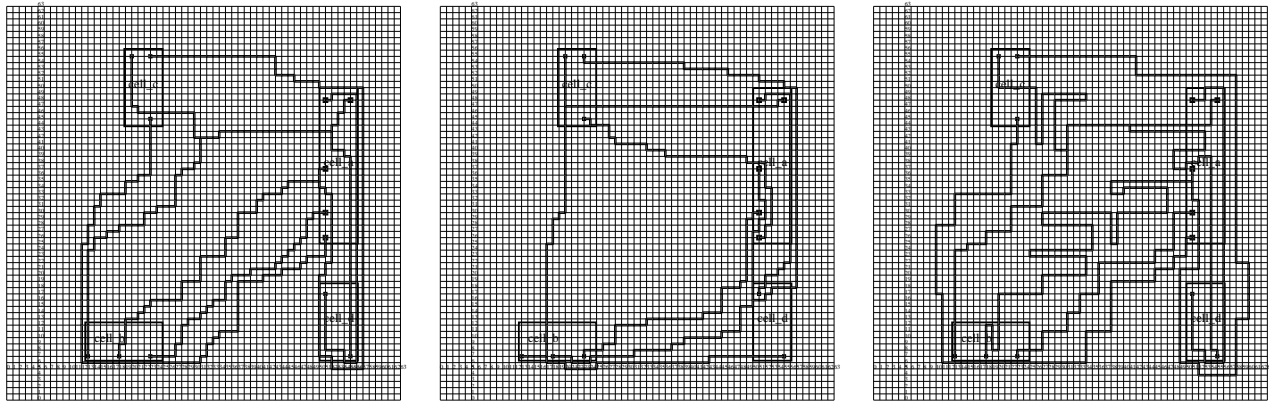


Fig. 7. Exp.B,C : Input trees of D3(left),Exp.B : length concerned(center) and Exp.C : weight ratio of 1:1:10(right)

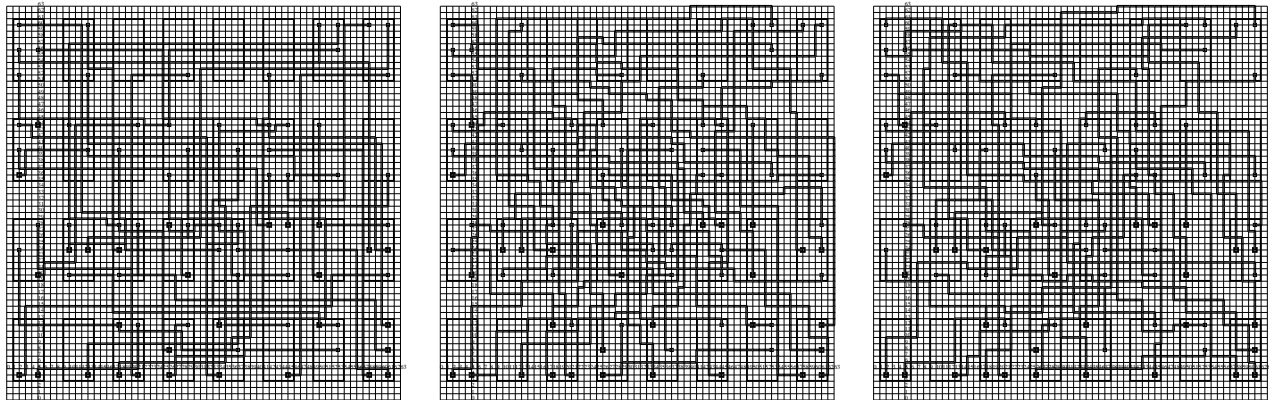


Fig. 8. Exp.D:Input trees(left), weight ratio of 1:9(center) and weight ratio of 1:3(right)

proposed the concepts of flip and dual-flip which are operations to transform the current route.

Implementing them to the SA controlled algorithm, we could show its adaptability to any performance evaluation. Since this paper is a proposal of the idea, the experiments were on the artificially made instances. For them, the experimental results showed a good performance.

Needless to say, the final objective of the work is to implement our improvement system to the practical system, where the evaluation function is so much complicated that the detailed evaluation of the performance, for example, delay of a line, can only be given by a look-up table[7], which our concept easily accepts.

ACKNOWLEDGMENTS

The authors are grateful to Associate professor Atsushi TAKAHASHI for his helpful discussion. This work is part of a project of CAD21 at Tokyo Institute of Technology.

REFERENCES

[1] H.Shin and A.Sangiovanni-V., "Mighty: A Rip-Up

and Reroute Detailed Router," in Proc. ICCAD-86, pp.2-5, 1986

[2] K.D.Boese, A.B.Kahng, B.A.McCoy, and G.Robins, "Rectilinear Steiner Trees with Minimum Elmore Delay" in Proc. 31th DAC, pp.381-386, 1994.

[3] M.Edahiro, "An Efficient Zero-Skew Routing Algorithm," in Proc. 31th DAC, pp.375-380, 1994.

[4] T.Gao and C.L.Liu, "Minimum Crosstalk Channel Routing," IEEE Trans. on CAD, vol.15, no.5, pp.465-474, 1996.

[5] A.Vittal and M.M.-Sadowska, "Crosstalk Reduction for VLSI," IEEE Trans. on CAD, vol.16, no.3, pp.290-298, 1997.

[6] B.Krauter and S.Mehrotra, "Layout Based Frequency Dependent Inductance and Resistance Extraction On-Chip Interconnect Timing Analysis," in Proc. 35th DAC. pp.303-308, 1998

[7] J.Lilis and P.Buch, "Table-Lookup Methods for Improved Performance-Driven Routing," in Proc. 35th DAC. pp.368-373, 1998